

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	0	(runtime adj register adj allocation) and @ad<"20020101"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/06/22 11:42
L2	1	(run-time adj register adj allocation) and @ad<"20020101"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/06/22 11:42
L3	8	(dynamic adj register adj allocation) and @ad<"20020101"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/06/22 11:42

Scholar

Results 1 - 10 of about 65 for "**dynamic optimization**" emulation . (0.14 seconds)

Dynamo: a transparent **dynamic optimization** system - group of 31 »

[All articles](#) [Recent articles](#)

V Bala, E Duesterwald, S Banerjia - Proceedings of the ACM SIGPLAN 2000 conference on ..., 2000 - portal.acm.org

... Dynamo is a **dynamic optimization** system (ie, the input is an executing native ... and it was more efficient to implement the interpreter by using **emulation**. ...

[Cited by 311](#) - [Web Search](#)

[PS] Design and implementation of a **dynamic optimization** framework for Windows - group of 6 »

D Bruening, E Duesterwald, S Amarasinghe - ... ACM Workshop on Feedback-Directed and **Dynamic Optimization** ..., 2001 - cag.lcs.mit.edu

... the code cache, one of the core components in any **dynamic optimization** system ... We chose this copy-paste approach over **emulation** because of the complexity of the ...

[Cited by 46](#) - [View as HTML](#) - [Web Search](#)

Transparent **dynamic optimization**: The design and implementation of Dynamo - group of 5 »

V Bala, E Duesterwald, S Banerjia - Hewlett Packard Laboratories Technical Report HPL-1999-78. ..., 1999 - hwswworld.com

... the execution of the program, whereas in the case of **dynamic optimization**, it is ...

These generate profile data during the initial run via **emulation**, and perform ...

[Cited by 60](#) - [View as HTML](#) - [Web Search](#)

Dynamic binary translation and optimization - group of 6 »

K Ebcioglu, E Altman, M Gschwind, S Sathaye - Computers, IEEE Transactions on, 2001 - ieeexplore.ieee.org

... Index TermsDDynamic compilation, binary translation, **dynamic optimization**, just-in-time ... executed memory operations do not incur **emulation** burden, whereas ...

[Cited by 36](#) - [Web Search](#) - [BL Direct](#)

Adaptive compression of syntax trees and iterative dynamic code optimization: Two basic technologies ... - group of 4 »

M Franz - Mobile Object Systems: Towards the Programmable Internet, 1997 - ics.uci.edu

... Fra94a, Fra94b], while the system architecture incorporating **dynamic optimization** is ongoing ... browser, a Telnet application with VT100 **emulation**, an electronic ...

[Cited by 30](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

Machine-adaptable dynamic binary translation - group of 7 »

D Ung, C Cifuentes - Proceedings of the ACM SIGPLAN workshop on Dynamic and ..., 2000 - portal.acm.org

... The process of mixing translation with **emulation** and runtime profiling brought about ...

to static ones, without taking advantage of **dynamic optimization** techniques ...

[Cited by 35](#) - [Web Search](#) - [BL Direct](#)

Optimization and precise exceptions in dynamic compilation

M Gschwind, E Altman - ACM SIGARCH Computer Architecture News, 2001 - portal.acm.org

... codes have been per- formed in previous architecture **emulation** systems, such as ... The DYNAMO **dynamic optimization** system per- forms **dynamic optimization** on HP-PA ...

[Cited by 10](#) - [Web Search](#) - [BL Direct](#)

Welcome to the opportunities of binary translation - group of 3 »

ER Altman, D Kaeli, Y Sheffer - Computer, 2000 - ieeexplore.ieee.org

... **Dynamic optimization** Sometimes a dynamic translator forms part of the translated code's execution thread, which means that ... **Emulation** Static translation ...

[Cited by 21](#) - [Web Search](#) - [BL Direct](#)

Optimization-based differential kinematic modeling exhibits a velocity-control strategy for dynamic ... - group of 6 »

X Zhang, AD Kuo, DB Chaffin - Journal of Biomechanics, 1998 - mie.uiuc.edu

... (1995), the implementation of **dynamic optimization**, in particular ... Further, the applicability of sequential static motion **emulation** is challenged by the fact ...
[Cited by 15](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

[Code Reordering and Speculation Support for **Dynamic Optimization** Systems - group of 13 »](#)
EM Nystrom, RD Barnes, MC Merten, WW Hwu - International Conference on Parallel Architectures and ..., 2001 - doi.ieeeecs.org
Page 1. Code Reordering and Speculation Support for **Dynamic Optimization**
Systems Erik ... 2.3 Other **Dynamic Optimization** Systems The ...
[Cited by 7](#) - [Web Search](#)

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 **Next**

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

Search Results

BROWSE

SEARCH

IEEE XPLORE GUIDE

SUPPORT

Results for "((dynamic optimization<and>emulation)) <and> (pyr >= 1951 <and> pyr <= 2001)"

Your search matched 9 of 1360403 documents.

A maximum of 250 results are displayed, 25 to a page, sorted by Relevance in Descending order.

 e-mail  printer friendly

» Search Options

[View Session History](#)

[New Search](#)

Modify Search

((dynamic optimization<and>emulation)) <and> (pyr >= 1951 <and> pyr <= 2001)

Search 

☐ Check to search only within this results set

Display Format: ☒ Citation ☐ Citation & Abstract

 **view selected items**

[Select All](#) [Deselect All](#)

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

- ☒ **1. Dynamic binary translation and optimization**
 Ebcioglu, K.; Altman, E.; Gschwind, M.; Sathaye, S.;
[Computers, IEEE Transactions on](#)
 Volume 50, Issue 6, June 2001 Page(s):529 - 548
 Digital Object Identifier 10.1109/12.931892
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(6164 KB) IEEE JNL
[Rights and Permissions](#)
- ☐ **2. Advances and future challenges in binary translation and optimization**
 Altman, E.R.; Ebcioglu, K.; Gschwind, M.; Sathaye, S.;
[Proceedings of the IEEE](#)
 Volume 89, Issue 11, Nov. 2001 Page(s):1710 - 1722
 Digital Object Identifier 10.1109/5.964447
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(232 KB) | Full Text: [HTML](#) IEEE JNL
[Rights and Permissions](#)
- ☐ **3. Control and coordination in hierarchical systems**
 Varaiya, P.;
[Proceedings of the IEEE](#)
 Volume 70, Issue 7, July 1982 Page(s):778 - 780
[AbstractPlus](#) | Full Text: [PDF](#)(440 KB) IEEE JNL
[Rights and Permissions](#)
- ☐ **4. Software-directed register deallocation for simultaneous multithreaded processors**
 Lo, J.L.; Parekh, S.S.; Eggers, S.J.; Levy, H.M.; Tullsen, D.M.;
[Parallel and Distributed Systems, IEEE Transactions on](#)
 Volume 10, Issue 9, Sept. 1999 Page(s):922 - 933
 Digital Object Identifier 10.1109/71.798316
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(1096 KB) IEEE JNL
[Rights and Permissions](#)
- ☐ **5. Welcome to the opportunities of binary translation**
 Altman, E.R.; Kaeli, D.; Sheffer, Y.;
[Computer](#)
 Volume 33, Issue 3, March 2000 Page(s):40 - 45
 Digital Object Identifier 10.1109/2.825694
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(647 KB) IEEE JNL
[Rights and Permissions](#)
- ☐ **6. An architectural framework for runtime optimization**
 Merten, M.C.; Trick, A.R.; Barnes, R.D.; Nystrom, E.M.; George, C.N.; Gyllenhaal, J.C.; Hwu, W.-M.W.;
[Computers, IEEE Transactions on](#)
 Volume 50, Issue 6, June 2001 Page(s):567 - 589

Digital Object Identifier 10.1109/12.931894

[AbstractPlus](#) | [References](#) | Full Text: [PDF\(3812 KB\)](#) IEEE JNL
[Rights and Permissions](#)



7. Continuous program optimization: Design and evaluation

Kistler, T.; Franz, M.;

[Computers, IEEE Transactions on](#)

Volume 50, Issue 6, June 2001 Page(s):549 - 566

Digital Object Identifier 10.1109/12.931893

[AbstractPlus](#) | [References](#) | Full Text: [PDF\(3856 KB\)](#) IEEE JNL
[Rights and Permissions](#)



8. Model predictive control of nonlinear systems: computational burden and stability

Chen, W.-H.; Ballance, D.J.; O'Reilly, J.;

[Control Theory and Applications, IEE Proceedings-](#)

Volume 147, Issue 4, July 2000 Page(s):387 - 394

Digital Object Identifier 10.1049/ip-cta:20000379

[AbstractPlus](#) | Full Text: [PDF\(752 KB\)](#) IEE JNL



9. Matching architecture and software technology for HPC systems

Vanneschi, M.;

[Parallel and Distributed Processing, 1999. PDP '99. Proceedings of the Seventh Euromicro Workshop on](#)

3-5 Feb. 1999 Page(s):2 - 9

Digital Object Identifier 10.1109/EMPDP.1999.746638

[AbstractPlus](#) | Full Text: [PDF\(32 KB\)](#) IEEE CNF
[Rights and Permissions](#)

[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)

© Copyright 2006 IEEE – All Rights Reserved

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published before January 2002

Terms used dynamic optimization emulation

Found 31 of 125,747

Sort results by

Display results


[Save results to a Binder](#)

[Search Tips](#)

[Open results in a new window](#)

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Results 1 - 20 of 31

Result page: [1](#) [2](#) [next](#)

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Dynamo: a transparent dynamic optimization system](#)



Vasanth Bala, Evelyn Duesterwald, Sanjeev Banerjia

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(156.03 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe the design and implementation of Dynamo, a software dynamic optimization system that is capable of transparently improving the performance of a native instruction stream as it executes on the processor. The input native instruction stream to Dynamo can be dynamically generated (by a JIT for example), or it can come from the execution of a statically compiled native binary. This paper evaluates the Dynamo system in the latter, more challenging situation, in order to emphasize the ...

2 [The benefits and costs of DyC's run-time optimizations](#)



Brian Grant, Markus Mock, Matthai Philipose, Craig Chambers, Susan J. Eggers

September 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.59 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

DyC selectively dynamically compiles programs during their execution, utilizing the run-time-computed values of variables and data structures to apply optimizations that are based on partial evaluation. The dynamic optimizations are preplanned at static compile time in order to reduce their run-time cost; we call this staging. DyC's staged optimizations include (1) an advanced binding-time analysis that supports polyvariant specialization (enabling both single-way and multi ...

Keywords: dynamic compilation, specialization

3 [Optimization and precise exceptions in dynamic compilation](#)



Michael Gschwind, Erik Altman

March 2001 **ACM SIGARCH Computer Architecture News**, Volume 29 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(508.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Maintaining precise exceptions is an important aspect of achieving full compatibility with a legacy architecture. While asynchronous exceptions can be deferred to an appropriate boundary in the code, synchronous exceptions must be taken when they occur. This introduces uncertainty into liveness analysis since processor state that is otherwise dead

may be exposed when an exception handler is invoked. Previous systems either had to sacrifice full compatibility to achieve more freedom to perform op ...

4 Software profiling for hot path prediction: less is more



Evelyn Duesterwald, Vasanth Bala

November 2000 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , Proceedings of the ninth international conference on Architectural support for programming languages and operating systems ASPLOS-IX**, Volume 34 , 28 Issue 5 , 5

Publisher: ACM Press

Full text available: [pdf\(286.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

5 Software profiling for hot path prediction: less is more



Evelyn Duesterwald, Vasanth Bala

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Publisher: ACM Press

Full text available: [pdf\(2.43 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

6 Machine-adaptable dynamic binary translation



David Ung, Cristina Cifuentes

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization DYNAMO '00**, Volume 35 Issue 7

Publisher: ACM Press

Full text available: [pdf\(1.23 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

Dynamic binary translation is the process of translating and optimizing executable code for one machine to another at runtime, while the program is "executing" on the target machine.

Dynamic translation techniques have normally been limited to two particular machines; a competitor's machine and the hardware manufacturer's machine. This research provides for a more general framework for dynamic translations, by providing a framework based on specifications of machines that ...

Keywords: binary translation, dynamic compilation, dynamic execution, emulation, interpretation

7 A hardware mechanism for dynamic extraction and relayout of program hot spots



Matthew C. Merten, Andrew R. Trick, Erik M. Nystrom, Ronald D. Barnes, Wen-mei W. Hmu

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2

Publisher: ACM Press

Full text available: [pdf\(320.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

This paper presents a new mechanism for collecting and deploying runtime optimized code. The code-collecting component resides in the instruction retirement stage and lays out hot execution paths to improve instruction fetch rate as well as enable further code optimization. The code deployment component uses an extension to the Branch Target Buffer to migrate execution into the new code without modifying the original code. No significant delay is added to the total execution of the program ...

8 Improving Java performance using hardware translation



Ramesh Radhakrishnan, Ravi Bhargava, Lizy K. John

June 2001 **Proceedings of the 15th international conference on Supercomputing**

Publisher: ACM Press

Full text available: pdf(254.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

State of the art Java Virtual Machines with Just-In-Time (JIT) compilers make use of advanced compiler techniques, run-time profiling and adaptive compilation to improve performance. However, these techniques for alleviating performance bottlenecks are more effective in long running workloads, such as server applications. Short running Java programs, or client workloads, spend a large fraction of their execution time in compilation instead of useful execution when run using JIT compilers. In ...

9 Novel ideas: Performance characterization of a hardware mechanism for dynamic optimization

Brian Fahs, Satarupa Bose, Matthew Crum, Brian Slechta, Francesco Spadini, Tony Tung, Sanjay J. Patel, Steven S. Lumetta

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available: pdf(1.31 MB) [Publisher Site](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We evaluate the rePLay microarchitecture as a means for reducing application execution time by facilitating dynamic optimization. The framework contains a programmable optimization engine coupled with a hardware-based recovery mechanism. The optimization engine enables the dynamic optimizer to run concurrently with program execution. The recovery mechanism enables the optimizer to make speculative optimizations without requiring recovery code. We demonstrate that a rePLay configuration performing ...

10 Increasing the size of atomic instruction blocks using control flow assertions



Sanjay J. Patel, Tony Tung, Satarupa Bose, Matthew M. Crum

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Publisher: ACM Press

Full text available: pdf(140.81 KB) ps(646.25 KB) [Publisher Site](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Overcoming the challenges to feedback-directed optimization (Keynote Talk)



Michael D. Smith

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization DYNAMO '00**, Volume 35 Issue 7

Publisher: ACM Press

Full text available: pdf(1.33 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Feedback-directed optimization (FDO) is a general term used to describe any technique that alters a program's execution based on tendencies observed in its present or past runs. This paper reviews the current state of affairs in FDO and discusses the challenges inhibiting further acceptance of these techniques. It also argues that current trends in

hardware and software technology have resulted in an execution environment where immutable executables and traditional static optimizations are ...

12 Profile-guided optimization across process boundaries



Erik Johansson, Sven-Olof Nyström

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization DYNAMO '00**, Volume 35 Issue 7

Publisher: ACM Press

Full text available: [pdf\(911.89 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe a profile-driven compiler optimization technique for *inter-process optimization*, which dynamically inlines the effects of sending messages. Profiling is used to find optimization opportunities, and to dynamically trigger recompilation and optimization at run-time. We apply the optimization technique on the concurrent programming language ERLANG, letting recompilation take place in a separate ERLANG process, and taking advantage of the facilities provided by ERLANG to dynami ...

13 Binary translation and architecture convergence issues for IBM system/390



Michael Gschwind, Kemal Ebcioglu, Erik Altman, Sumedh Sathaye

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Publisher: ACM Press

Full text available: [pdf\(1.44 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities which can be exploited by binary translation software ...

14 An evaluation of staged run-time optimizations in DyC



Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, Susan J. Eggers

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation PLDI '99**, Volume 34 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.54 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Previous selective dynamic compilation systems have demonstrated that dynamic compilation can achieve performance improvements at low cost on small kernels, but they have had difficulty scaling to larger programs. To overcome this limitation, we developed DyC, a selective dynamic compilation system that includes more sophisticated and flexible analyses and transformations. DyC is able to achieve good performance improvements on programs that are much larger and more complex than the kernels. We ...

15 Techniques for obtaining high performance in Java programs



Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja

September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Publisher: ACM Press

Full text available: [pdf\(816.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

Keywords: Java, Java virtual machine, bytecode-to-source translators, direct compilers, dynamic compilation, interpreters, just-in-time compilers

16 Adaptive optimization in the Jalapeño JVM



Matthew Arnold, Stephen Fink, David Grove, Michael Hind, Peter F. Sweeney

October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '00**, Volume 35 Issue 10

Publisher: ACM Press

Full text available: [pdf\(716.90 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Future high-performance virtual machines will improve performance through sophisticated online feedback-directed optimizations. This paper presents the architecture of the Jalapeño Adaptive Optimization System, a system to support leading-edge virtual machine technology and enable ongoing research on online feedback-directed optimizations. We describe the extensible system architecture, based on a federation of threads with asynchronous communication. We present an implementation of t ...

17 Partial evaluation as a means for inferencing data structures in an applicative language: a theory and implementation in the case of prolog



H. Jan Komorowski

January 1982 **Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Publisher: ACM Press

Full text available: [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

An operational semantics of the Prolog programming language is introduced. Meta-IV is used to specify the semantics. One purpose of the work is to provide a **specification of an implementation** of a Prolog interpreter. Another one is an application of this specification to a formal description of program optimization techniques based on the principle of **partial evaluation**. Transformations which account for pruning, forward data structure propagation and opening (which al ...

18 On the development of a site selection optimizer for distributed and parallel database systems



Fotis Barlos, Ophir Frieder

December 1993 **Proceedings of the second international conference on Information and knowledge management**

Publisher: ACM Press

Full text available: [pdf\(1.11 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

19 DSL implementation using staging and monads



Tim Sheard, Zine-el-abidine Benaissa, Emir Pasalic

December 1999 **ACM SIGPLAN Notices , Proceedings of the 2nd conference on Domain-specific languages PLAN '99**, Volume 35 Issue 1

Publisher: ACM Press

Full text available: [pdf\(923.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The impact of Domain Specific Languages (DSLs) on software design is considerable. They allow programs to be more concise than equivalent programs written in a high-level programming languages. They relieve programmers from making decisions about data-structure and algorithm design, and thus allows solutions to be constructed quickly. Because DSL's are at a higher level of abstraction they are easier to maintain and reason about than equivalent programs written in a high-level language, and ...

20 Optimising hot paths in a dynamic binary translator



David Ung, Cristina Cifuentes

March 2001 **ACM SIGARCH Computer Architecture News**, Volume 29 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(890.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In dynamic binary translation, code is translated "on the fly" at run-time, while the user perceives ordinary execution of the program on the target machine. Code fragments that are frequently executed follow the same sequence of flow control over a period of time. These fragments form a hot path and are optimised to improve the overall performance of the program. Multiple hot paths may also exist in programs. A program may choose to execute in one hot path for some time, but later switch to another ...

Keywords: binary translation, dynamic compilation, dynamic execution, run-time profiling

Results 1 - 20 of 31

Result page: [1](#) [2](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)